AD 653640

AD

DYNAMIC ADAPTIVE DATA BASE MANAGEMENT STUDY

Third Quarterly Progress Repo..

16 November 1966—15 February 1967

LINGUISTICS RESEARCH CENTER
The University of Texas at Austin
Box 7247, University Station
Austin, Texas 78712

ECOM

UNITED STATES ARMY ELECTRONICS COMMAND · FORT MONMOUTH, N.J.

31

DYNAMIC ADAPTIVE DATA BASE MANAGEMENT STUDY

16 November 1966 - 15 February 1967

W.P. Lehmann, Director

E. D. Pendergraft, Associate Director

prepared for

U. S. Army Electronics Command
Fort Monmouth, New Jersey, 07703
Contract No. DA 28-043 AMC-02276 (E)
File No. 28-043-J6-22395 (E)

LINGUISTICS RESEARCH CENTER
The University of Texas at Austin
Box 7247, University Station
Austin, Texas 78712

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Technical Report
ECOM-02276-3

May 1967

CONTENTS

ABSTRACT

The feasibility of dynamic adaptive data base management is being investigated using linguistically specified processors, automatic classification based on clumping theory, and a monitoring system capable of exercising judgment concerning the need for automatic revision of the data base.

Preceding Page Blank

1 PURPOSE

The purpose of the investigation under this contract [DA 28-043 AMC 02276(E)] is the development of a body of theory regarding the nature and function of classification systems and the organization of data structures necessary for effective, adaptive data base management. Research concerns the formation and manipulation of extremely large and dynamically changing data bases, including the feasibility of automatic revision by a monitoring system capable of exercising judgment as to when revision processors should be invoked.

2 CONFERENCES

Weekly conferences continued to be held to discuss the theoretical issue under investigation. Individual members of our group also kept in touch with research in automatic classification being done in connection with academic programs on the campus. Major topics were:

(a) limitations of automatic classification procedures,

(b) an evolutionary approach to the development of a general adaptive capability,

(c) specialization of the general adaptive capability to the specific requirements of data management.

3 RESEARCH SUMMARY

The objectives of this investigation are being approached from two points of view, which might be contrasted as the theoretical-deductive and the experimental-inductive. At the beginning of the study these two research objectives were not adequately distinguished, nor did it seem important to do so.

The design and implementation of an experimental programming system was immediately undertaken with an aim toward getting experiments in adaption under way early enough to produce useful results during the contract period. Some of the techniques we proposed to implement mechanically had been sketched in earlier studies [1]. Consequently, system analysis and techniques of programming were stressed in the first quarter. Theoretical discussions were concerned mainly with working out the details of an existent conceptual framework.

The second quarter marked a turning point in our thinking. Through theoretical extensions, we had elaborated the original scheme to include a number of new technical possibilities. At the same time the underlying assumptions

were simplified, so that these new possibilities could be accounted for as special cases within a more general theory.

Less tangible, but we believe more important, was our renewed and enlarged appreciation of the method of "evolutionary programming" [2] as an approach to the development of complex information systems. Let us present this concept, as we now understand it, before discussing its implications for our investigation.

3.1 Evolutionary Programming

We take it for granted that schemes for data base management should be evaluated in terms of the facilities they provide for (a) getting knowledge into an information system, and (b) for using that knowledge intelligently. Theories of data management are thus inextricable from theories of knowledge and intelligence.

Further, students of human behavior have argued convincingly that there is an intimate connection between knowing and intelligent doing [3]. Because our largess of knowledge has been perpetuated by writing, we are prone to thinking that our information needs can be satisfied by merely storing books, abstracts, messages or accounting

entries in a computer. But after establishing such a data base, there remains the formidable task of analyzing and classifying its information content so that an intelligent human or machine can use it.

What must pass for knowledge in an intelligent machine, accordingly, is the compendium of data which results from something happening to information _after_ it is stored in the data base. And what must pass for intelligence in a knowledgeable machine is the repertoire of processes addressed to that compendium, not to the data base itself unless trivially, and applying the machine's knowledge to some useful purpose.

There is the complication that an intelligent machine may be knowledgeable about processes, so that descriptions of processes can be a type of data in its store of knowledge. As a corollary, the machine may know about the processes by which it acquires knowledge and by which it uses knowledge intelligently. These two types of processes are said to underlie mechanical _learning_ and _performance_, respectively. The machine's compendium of knowledge, in addition, may contain descriptions of processes realizable by entities other than itself.

We assume that a necessary condition for versatility in an information system, though obviously not a sufficient one, is complexity in the data that constitute its knowledge and in the processes that underlie its intelligence. This view is supported tenuously by biological analogies, yet it seems the best rationalization for our present ineptitude in constructing versatile information systems. It may have additional value as an indicator that our difficulties are methodological as well as theoretical.

As system complexity increases, we find undeniably that no amount of theoretical contemplation can bring to light all technical issues which must be faced before commitments are made to implementation. Here is an embarrassing dilemna. Like sailors on the edge of an uncharted sea, we may sail confidently ahead or turn back in humiliation. Ahead lie the shoals of debugging, redesign and revision. If pressed, we accept the risk and hope for the best. But backing out can be explained in the name of prudence.

Theoretical-deductive methods are, accordingly, aids to navigation for information system development.

They can be used as means of demonstrating (proving) with certainty or heuristically that a proposed system design is likely to succeed in its purpose (feasible) and is worthy of implementation (useful). When ordered under full sail, however, we must insist on accurate charts. In that predicament it seems best to be practical (timid) in proposing mechanizable theories of knowledge and intelligence.

Another methodological decision on the side of prudence is that, in the symbiosis between man and machine, the machine should be on the receiving end of knowledge and man on the receiving end of intelligence. According to this notion, then, machines should be passive recipients of knowledge created by a superior (human) experimental-inductive capability. Having communicated his own knowledge to machine, man should have every right to expect a mechanical performance mirroring his own intelligence.

Modeling man in the computer has not worked out very well in practice. Two decades spent in describing a meager part of English for consumption by machines is commendable, though not reassuring. Under hard scrutiny, our egocentric insistence that we must hand-fabricate the

the machine's knowledge in order to be the recipients of our own intelligence does not seem realistic.

In short, our current methodological position is based on the assumption that machines are incapable of applying experimental-inductive methods toward acquiring knowledge for themselves as well as men can describe knowledge to them. This proposition needs more argument than a boastful comparison of human and mechanical inductive capabilities.

Evolutionary programming is another avenue to knowledgeable and intelligent machines. As Fogel, Owens and Walsh point out in the book already cited, it is an attempt to model evolution rather than man. And, unlike bionic and heuristic programming approaches which attempt to describe man as he exists in nature, evolutionary programming is primarily normative. It is an attempt to program evolutionary processes as they might occur in nature; to describe what ought to be rather than what is. It attempts to create knowledgeable and intelligent systems as a product of "fast time" simulation of evolutionary processes.

Knowledgeable and intelligent systems resulting from evolutionary programming cannot be expected, ipso facto, to

resemble man. Their usefulness might indeed be evaluated in terms of their ability to solve problems which have resisted solution by mankind -- as Fogel and his associates recommend. Or machines, having access to their environment through mechanical sensors or effectors, may become more sensitive or responsive than man -- or merely sensitive and responsive in ways different from, yet valuable to, man.

We shall not resist these audacious recommendations, since evolutionary programming appears to offer a way out of our dilemma. It cautions us to reduce sail, to proceed methodically across shoals only poorly charted. If we are to be the designers and implementers of exceptional knowledge and intelligence, then we must play the evolutionary game, step by step through successive stages of knowing and intelligent doing, albeit in "fast time."

Once decided to act like gods instead of mechanics, we may put our efforts into hastening and guiding the evolution of complex machines rather than describing every nut and bolt in them. Secondly, it should be kept in mind that the decision to guide the evolutionary development of complex systems is not incompatible with the goal of creating mechanical systems which are functionally, though not structurally, like man.

Evolutionary programming explicitly assumes that mechanized knowledge and intelligence must evolve together toward greater complexity and versatility. At each stage of development, the information system will be required to demonstrate three characteristics (explained in general terms by Ashby [4]):

(a) stability, in that the system's dynamically changing compendium of knowledge will have come to contain (for practical purposes) all of the data inducible from its history of performance by means of its current learning capability,

(b) ultrastability, in that stability will have been attained with acceptable values for all critical variables (directly or indirectly) evaluating its current capacity for performance based deductively on that knowledge, and

(c) polystability, in that only (minor) parts of that knowledge would be modified should the value of some critical parameter be outside of its prescribed range, causing the system to readapt.

Descriptive adaptation, initiated by some basic restructuring of the system's environment, will be

distinguished from theoretical adaptation, initiated by some basic restructuring of the system itself. Both types of adaptation lead to alterations of the data constituting knowledge. However, only theoretical adaptation entails alteration of the processes underlying learning or performance, and hence alteration of the very conditions under which descriptive adaptation is to take place.

The decision to embrace evolutionary programming as an approach to the development of complex information systems, therefore, is tantamount to the decision to exercise control over theoretical instead of descriptive adaptation.

Controlling what the system ought to be, instead of what it ought to know, seems an appropriate pursuit for designers and implementers. For very complex systems, this is the sole choice possible to us in terms of sheer labor. And, were we to succeed in the doubtful task of hand-working the knowledge of a complex system, we would also saddle ourselves with periodic readjustments to affect descriptive adaptation. Thus a poor rate of descriptive adaptation would undoubtedly limit applications.

Mechanical learning is consequently to be compared with a questionable human performance in describing knowledge to machines, as often as not the error-prone result of boredom. Certainly the job is unrewarding for man's vaunted experimental-inductive abilities. The comparison, furthermore, is not essentially between the experimental-inductive abilities of men and machines in direct confrontation with nature. What will chiefly concern us is a restricted kind of mechanical learning aimed at acquiring a restricted kind of knowledge -- the knowledge needed to break the various symbolic codes which convey information in the data base.

This finding, in the second quarter of the contract period, brought us to concentrate our experimental efforts in the area of linguistic adaptation. Our theoretical efforts, in contrast, were concentrated on the task of describing, as well as we can see ahead from our present position, the characteristics which a knowledgeable and intelligent information system ought to have.

As a result, we came to look upon our theoretical-deductive investigations as a means of planning the evolutionary stages through which our system must pass in the progression toward complexity and versatility. Experimental-inductive

investigations have been oriented to the down-to-earth business of taking a first step in a planned series of theoretical adaptations.

Progress in these two technical areas will now be summarized.

3.2 Self-organizing Semiotic System

In brief, the system under theoretical investigation is to be machine independent except for elementary, purposive (goal-directed) processors which will couple it to its environment. The processors will, in effect, convey all stimuli and responses across the system's internal and external boundaries. All other linkages, determining a hierarchy of purposive processors, will be described by operational rules. These, in conjunction with descriptions of the processes to be realized by the processors, will constitute a formal description of the system's total operating capability, specifying how complex processors are to be constructed from the elementary ones.

Rules will be of two types, exemplified by:

$$a \varepsilon P_1, \; b \varepsilon P_2, \; c \varepsilon P_3$$

$$((P_3 \frown (P_1 \wedge \sim P_2)) \vee (P_2 \frown P_1) \rightarrow P_4$$

Here a, b and c are purposive processes (or processors) having the goals P_1, P_2 and P_3, respectively. P_4 is a goal which may be attained by means of a complex process (or processor) whose subprocesses (or subprocessors) include a, b and c. Logical symbols are to be interpreted in terms of success or failure; thus, the complex processor will successfully attain the goal P_4 if P_3 and then P_1 but not P_2 are successfully attained, or if P_2 and then P_1 are attained. Negation is seen to turn success into failure and vice versa.

Elementary processors may have parameters. In consequence, the parameters of a complex processor are those of its elementary subprocessors. Alternative processors may realize the same elementary process; the parameters of that process are those of all processors realizing it. Alternative processes may attain the same goal; in other words, various means may be directed toward some common end. The parameters of the process seeking to attain that end are those of all processes which may be means to it.

A <u>deductive</u> processor will interpret the rules in order to actively seek to attain a given goal. Using the "knowledge" constituted by the collection of rules,

and perhaps other data summarizing information about the environment, this processor will attempt to make "intelligent" choices among the alternative processes or processors which might be useful in gaining that objective. Analogues of thinking -- that is to say, planning ahead, estimating the probable outcomes of possible courses of action, determining ethical constraints, and so on -- will be processes realized by processors of this kind.

A control processor, designed to execute processor description in which there are no remaining alternatives, will actually carry out such explicit courses of action as are specified to it. One should note the submission of a course of action to this control processor will be the mechanical analogy of an organism's decision to act. Nevertheless, having made its selection, the deductive processor need not submit its entire plan of action to the control processor for execution. Part of it may be held back as tentative. As much of the plan will be executed as seems likely to succeed, or likely to gain additional information about the environment as a basis for more precise planning.

These deductive acts, whether overt or merely planned and evaluated with regard to probable outcome, will

be oriented to an environment conceptually external to the system proper. The system will be thought of as performing in that environment and modifying it structurally by every overt action; the whole system will be one of the objects in that environment.

Probabilities will be associated with the individual rules. Those of all means to a given end will sum to one. The probability of an explicit course of action will be the product of the probabilities of the individual rules describing it. Thus, deductive choice-making as described above is to be conceptualized an independent stochastic process [1].

An inductive processor will review patterns of success or failure in the system's performances, or other summary information about the system proper as an environment, as a basis for learning. In a word, inductive acts are to be turned inward instead of outward, toward the goal of modifying the rules or their probabilities, or other elements of the system's knowledge. And, because a part of that knowledge can be descriptions of these deductive or inductive processors, induction may alter the system's internal structure or mode of functioning.

For the system to be a versatile performer in its external environment, it should have a wide range of alternatives on which to exercise its deductive choice-making abilities. Versatile learning, as we are already finding experimentally, also requires a wido range of alternatives to be placed at the disposal of the inductive processor. Hence, the desideratum of learning is a versatile performance in the internal environment provided by the system itself.

Some of the elementary processors are to be <u>afferent</u> in that their successes or failures will merely transmit information about the (internal or external) environment without modifying it. The rest are to be <u>efferent</u> processors in that their successes will signal the completion of some particular modification of the environment, or their failures the frustration of some attempt to modify it. Successful execution of an afferent processor will therefore signal the presence, and failure the absence, of certain stimuli. The success or failure of an efferent processor will signal whether or not certain responses came off as intended.

Mediating stimuli and responses across the external boundaries of a knowledgeable and intelligent system will be

the process which Morris [5] calls <u>semiosis</u>, in which something is a sign to some organism.

<u>Mechanical semiosis</u>, the analogue of organic semiosis in which something is a sign to some mechanism, is presumed in our hypothesis to include the following subprocesses:

(a) perception of complex stimuli that evidence either linguistic or non-linguistic features (signs) in an environment,

(b) conceptualization of complex features that evidence meanings (significata),

(c) symbolization of complex meanings (references or designata) that evidence other complex meanings (referents or denotata), and

(d) valuation of complex meanings, presented to the system either directly through conceptualization or indirectly through symbolization, that evidence initiators or parameter values (interpretants) of particular processes or processors (interpreters) which are purposive in that their performances must succeed or fail.

A system capable of performing mechanical semiosis for purposes of communication or control is called a <u>semiotic</u> system.

Fundamentally, the performance of a semiotic system is taken to be a transition from stimuli -- through perception, through conceptualization and perhaps symbolization and through valuation -- to responses. Machines more versatile than those now available would result from improved modes of perception or valuation alone. Or a machine might conceptualiz e without being able to symbolize. We believe, however, that devices without all of these capabilities would have severe limitations when applied to data base management or other complicated tasks.

A semiotic system should be expected to evolve through successive stages of internal structuring. This developmental process is to be "theoretically" adaptive in that, at each of the stages, the modified or extended system will be required to readapt "descriptively" to its environment. Thus, the environment is to be accepted as the final arbiter of mechanical systems as well as organic. What will be gained is an empirical test of success or failure for <u>each</u> system modification or extension, so that costly mistakes can be minimized.

Our attack should start with perceptual performance -- the process most intimately in touch with the external environment -- and with its internal counterpart, perceptual learning. It may then radiate along the course of semiosis though conceptual, symbolic and valuative performance and learning. However, this should not lead us to conclude that we must fully understand each subprocess of semiosis before going on to the next.

We observe that evolutionary programming should tend toward polystability in knowledge <u>about</u> the structural requirements of semiotic systems. If Ashby's arguments are valid, then neglect of this principle may be one cause of our misfortunes in trying to cope with complex theories. But achieving a state of knowledge in which relatively small corrections would permit us to readapt theoretically does not necessarily imply that the corrections must be made at the perimeter of our understanding.

The chief cause of our difficulties, we presume, is a tendency to approach extreme complexity without a definite plan or, worse still, with a plan looking no farther ahead than is "practical." While evolution

theorists assure us that even a helter-skelter attack can be creative in the long run, the cost is clearly prohibitive for information system development. Shortsighted, we are forever bumping against reality. The monstrosities we create have little chance of survival and we are not sad to see them go.

We cannot force nature; nor can we afford to wait. Planning the evolution of complex systems means predicting and fostering innumerable, small opportunities and taking advantage of them as soon as they arise. This simple strategy will give us maximum progress. If every extension must be justified on practical as well as theoretical grounds, we will not go very far. The interests of theory and practice may coincide occasionally, but not often enough to construct complex systems in "fast time."

With this broad plan before us, therefore, we turn to the particular opportunities which are the focus of our experimental-inductive investigations.

3.3 Syntactic Self-organization

Syntactics, semantics, and pragmatics are the traditional divisions of semiotic, the study of sign-processes. Syntactics investigates relations among signs, without regard to their meaning or import. When significata, but not interpretants or interpreters, are among the relata under study, the investigation is one of semantics. Pragmatics studies relations involving interpretants or interpreters [5].

The analogues of perception which can be based on syntactic relations are our current experimental interest. Conceptualization and symbolization are based semantically within our theories, and valuation presupposes pragmatic relations. Feedback from these higher subprocesses of semiosis are expected to play an important role in perception at a later stage of inquiry.

Syntactic self-organization within a restricted linguistic environment is our present experimental goal.

In particular, the machine has been provided with the elementary processors needed to recognize the individual letters, numerals, and punctuation which transcribe a machine-related corpus of contemporary English [4] -- about 20,000

running words of newspaper copy discussing politics. Concatenation between these individual symbols, or strings of them, is the only logical relation of which the machine has knowledge.

Successive cycles of deduction and induction are carried out. In the first cycle, the mechanical system has no other knowledge than the above information about individual symbols. Automatic syntactic analysis is performed by deductive processors which apply the current syntactic description (in the first cycle the trivial one) to about 2,000 running words of the corpus. Inductive processors then review patterns of success or failure, make inference about syntactic modifications or extensions (in some cases using automatic classification) and prepare a revised syntactic description for use in the next cycle. When the end of the corpus is reached, another pass through it is begun.

The machine's knowledge, consequently, changes dynamically as it learns more about its environment. Syntactic self-organization is to be regarded as successful if the description stabilizes in the sense described above.

Two trial runs have been made and a third experiment is in progress. No attempt was made to control the system in the first two experiments. All parameters were set to minimum values, so that induction would progress as rapidly as

possible for the purpose of testing programs. After five or six cycles, both descriptions became "unstable" in that they could no longer be used economically by the deductive processors.

The experiment now in progress is being controlled through manual manipulation of the control parameters. Through the experience gained we hope to devise the control strategies required for ultrastable syntactic systems.

4 PROGRESS IN QUARTER

A good part of our time during the quarter was spent in managing our experiment. By exercising control through various parameters we have thus far been able to keep the syntactic description from going unstable, either through proliferation of too many rules, or rules which are too general.

As one might anticipate, we have found that syntactic descriptions coded by the machine differ from those coded by linguists. Although many of the programs involved in our experiment have been in use for some time [1], the machine-coded data apparently created various processing conditions which had not been encountered in the manually-coded. Delays were therefore incurred as dormant programming errors were repaired.

It also became necessary to make several changes in the inductive processors already in operation and to stress programming of those planned but not implemented. As a consequence of these more urgent needs, proposed experiments in automatic classification were postponed to the last quarter. Improvements were made, however, in the classification algorithms being employed in syntactic adaptation.

In all, progress was more than satisfactory. We have tried to exhibit some of the refinements in our objectives which have resulted from theoretical discussions. Technical conclusions, both theoretical and experimental, will be reserved for our final report because it seems appropriate that they should appear together.

5 CONCLUSIONS

Evolutionary programming appears to be a promising methodological innovation. Although its premises are not flattering, one should be ill advised to reject this approach out of hand. The only complex information systems now in existence were developed by similar methods. Humans may not be equal to the task of creating complex systems from scratch and to order. They, however, may be able to control evolutionary processes well enough for complex systems to evolve in "fast time" roughly to specifications.

6 PLANNING FOR THE NEXT QUARTER

During the next quarter we will concentrate on getting a maximum return from our experiment in syntactic self-organization. The classification experiments delayed in this quarter will also be performed. Theoretical work will be brought to conclusion in the middle of the quarter so that we can organize our conclusions for the final report.

REFERENCES

1. W.P. Lehmann and E.D. Pendergraft, "Linguistic Information Processing Study, Final Report, Austin: Linguistics Research Center, 1967.

2. L.J. Fogel, A.J. Owens, and M.J. Walsh, Artificial Intelligence Through Simulated Evolution, New York: John Wiley & Sons, 1966.

3. D.O. Hebb, Organization of Behavior, New York: Science Editions, 1961.

4. R.W. Ashby, Design For a Brain, New York: John Wiley & Sons, 1960.

5. C. Morris, Signs, Language and Behavior, New York: Prentice Hall, 1946.

6. W.N. Francis, "Manual of Information to Accompany a Standard Sample of Present-Day Edited American English, For Use With Digital Computers," Providence: Brown University, 1964.

PERSONNEL

The following personnel are reported under this contract. Figures give total hours worked.

L. D. Childress	(80)
B. C. Foster	(236)
N. F. Hirst	(418)
E. D. Pendergraft	(236)
C. E. Perkins	(412)
J. K. Schieffer	(444)
E. P. Shaw	(108)
J. C. Smith	(566)

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Linguistics Research Center University of Texas Austin, Texas 78712	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP N/A

3. REPORT TITLE

Dynamic Adaptive Data Base Management Study

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Third Quarterly Report

5. AUTHOR(S) (Last name, first name, initial)

Lehmann, Winfred P.
Pendergraft, Eugene D.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
May, 1967	28	6

8a. CONTRACT OR GRANT NO. DA 28-043-AMC-02276(E) b. PROJECT NO. 1P6 20501 A485 c. Task No. 03 d. Sub-task No. 02	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) ECOM-02276-3

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	US Army Electronics Command Fort Monmouth, NJ (AMSEL-NL-P-1)

13. ABSTRACT

The feasibility of dynamic adaptive data base management is being investigated using linguistically specified processors, automatic classification based on clumping theory, and a monitoring system capable of exercising judgment concerning the need for automatic revision of the data base.

DD FORM 1473

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Data Base Management						
Adaptive Data Management						
Automatic Classification						
Clumping Theory						
Automatic Data Revision						
Syntactic Self-Organization						
Information Retrieval						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.